

Final Evaluation: 40%

|  |  |
| --- | --- |
| **Course Identification** | |
|  |  |
| Name of program – Code: | COMPUTER SCIENCE TECHNOLOGY – VIDEO GAME PROGRAMMING – 420.BX |
| Course title: | **INFORMATION SYSTEM AND PROJECT METHODOLOGY II** |
| Course number: | 420-J12-AS |
| Group: | 00160 |
| Teacher’s name: | ADIN ASHBY |
| Duration: | 3 periods (150 minutes) |
| Semester: | Winter 2025 |
| **Student Identification** | |
| Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Student number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_  Date: May 5, 2025 Result: \_\_\_\_\_\_\_\_\_\_\_\_\_\_  I declare that this is an original work, and that I credited all content sources of which I am not the author (online and printed, images, graphics, films, etc.), in the required quotation and citation style for this work. | |
| **Standard of the Evaluated Competency(ies)** | |
|  | |

**Statement of the evaluated competency(ies) – Code(s)**

1. Develop data exchange services -00SV
2. Collaborate on the design of applications –00SY

|  |
| --- |
| **Instructions** |
|  |
| The use of AI and internet are not allowed during the exam. However, students are allowed to use their class notes only.No break is allowed during this exam. Students are not allowed to exit the examination room before half of the allotted time has passed. Once a student has exited the classroom, he/she may not re-enter (IPEL – Article 5.12.4).The teacher will not answer questions during the exam.Students must remain silent during the exam.It is the teacher’s responsibility to identify language errors. If such errors are found, teachers may apply a penalty of up to 20% of the grade (IPEL – Article 5.7).Plagiarism, attempts at plagiarism or complicity in plagiarism during a summative evaluation results in a mark of zero (0). In the case of recidivism, in the same course or in another course, the student will be given a grade of '0' for the course in question.(IPEL – Article 5.16)  * Students are responsible to make sure they have their login credentials to submit their solution. * Students are responsible to make sure they have submitted the correct version of their exam. |
| |  | | --- | | **Mark Breakdown** | |  |   This evaluation is on 100 points, distributed as follows:   |  |  |  | | --- | --- | --- | | **Question 1** | Implement a Game Asset Management System | For a total of 50 points | | **Question 2** | Implement a Character Customization System | For a total of 50 points | | **TOTAL: 100 POINTS** | | |   **Instructions**  **Question 1**  Implement a Game Asset Management System  **Problem Statement**  You are tasked with developing a **Game Asset Management System** that efficiently loads, caches, and provides access to game assets such as textures, models, and sound files. Your implementation should incorporate at least **two design patterns** from the following list:   * **Flyweight**: Optimize memory usage by sharing common assets among different game objects. * **Facade**: Provide a simplified interface for asset loading and retrieval. * **Proxy**: Implement lazy loading of assets, loading them only when needed. * **Bridge**: Separate abstraction (asset types) from implementation (asset storage methods).   **Requirements**   * Implement at least **two design patterns** from the above list. * Write a short report (a page or less) explaining how the design patterns were applied and their benefits. * Your implementation should be **object-oriented** and well-structured. * Ensure the code is **modular** and follows **SOLID principles**.   **Question 2**  Implement a Character Customization System  **Problem Statement**  You are required to develop a **Character Customization System** that allows players to customize their character’s appearance, attributes, and abilities. The system should support flexible object creation and allow modifications to the character without modifying its core structure. Your implementation should incorporate at least **two design patterns** from the following list:   * **Prototype**: Enable cloning of character templates to quickly create new characters with predefined attributes. * **Builder**: Provide a structured way to construct a character step by step. * **Factory Method**: Encapsulate character creation logic for different character classes (e.g., Warrior, Mage, Rogue). * **Bridge**: Decouple character abstraction (appearance, abilities) from its implementation (different storage formats or rendering engines).   **Requirements**   * Implement at least **two design patterns** from the above list. * Write a short report (a page or less) explaining how the design patterns were applied and their benefits. * Your implementation should be **object-oriented** and well-structured. * Ensure the code is **modular** and follows **SOLID principles**.  |  | | --- | | Evaluation grid | | **Questions 1 (50 points)**   |  |  | | --- | --- | | **Element of competency:** Develop data exchange services – 00SV | | | **Performance criteria** | **weight** | | 1.1 Accurate analysis of design documents | **/5** | | 1.2 Proper identification of the tasks to be carried out | **/5** | | **Element of competency: Collaborate on the design of applications – 00SY** | | | 2.1 Appropriateness of the recommendations regarding the choice of software architecture | **/10** | | 3.4 Object-oriented modelling compliant with principles of encapsulation, inheritance, composition and polymorphism | **/10** | | 3.5 Proper choice or production of algorithms | **/15** | | 3.6 Compliance with nomenclature rules | **/2.5** | | 4.5 Compliance with application development standards, methods and best practices | **/2.5** | | |
|  |
| **Questions 2 (50 points)** |

|  |  |
| --- | --- |
| **Element of competency:** Develop data exchange services – 00SV | |
| **Performance criteria** | **weight** |
| 1.1 Accurate analysis of design documents | **/5** |
| 1.2 Proper identification of the tasks to be carried out | **/5** |
| **Element of competency: Collaborate on the design of applications – 00SY** | |
| 2.1 Appropriateness of the recommendations regarding the choice of software architecture | **/10** |
| 3.4 Object-oriented modelling compliant with principles of encapsulation, inheritance, composition and polymorphism | **/10** |
| 3.5 Proper choice or production of algorithms | **/15** |
| 3.6 Compliance with nomenclature rules | **/2.5** |
| 4.5 Compliance with application development standards, methods and best practices | **/2.5** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CORRECTION GRID FOR LANGUAGE**   |  |  |  |  | | --- | --- | --- | --- | | Clear Communication | Clear Communication, **most of the time** | Vague Communication | Unclear Communication | | - 0 | - 0,5 | - 1,5 | - 2 | | (Word Choice)  Use of precise and rich vocabulary | (Word Choice)  Use of precise vocabulary | (Word Choice)  Use of imprecise vocabulary | (Word Choice)  Use of inappropriate vocabulary | | - 0 | - 0,5 | - 1,5 | - 2 | | (Format/Type of work)  Respect of norms | (Format/Type of work)  Respect of **most of the** norms | (Format/Type of work)  Non-respect of the norms | (Format/Type of work)  Inappropriate in relation to the required norms | | - 0 | - 0,5 | - 1,5 | - 2 | | (Linguistic Code)  (≤2 mistakes / page) | (Linguistic Code)  (3-7 mistakes/page) | (Linguistic Code)  (8-10 mistakes/ page) | (Linguistic Code)  (>10 mistakes/  page) | | - 0 | - 0,5 - 2.5 | - 2.5 - 3.5 | - 4 | |